Spring MVC - 파일 업로드/다운로드

Multipart?

- 웹 클라이언트가 요청을 보낼 때 http프로토콜의 바디 부분 에 데이터를 여러 부분으로 나 눠서 보내는 것
- 보통 파일을 전송할 때 사용한 다.

Subject: Formatted text mail MIME-Version: 1.0 Content-Type: multipart/alternative; boundary=boundary42 --boundary42 Content-Type: text/plain; charset=us-ascii Body part 1 ... plain text version of message goes here ... --boundarv42 Content-Type: text/enriched Body part 2 ... RFC 1896 text/enriched version of same message aoes here ... --boundary42 Content-Type: application/x-whatever Body part 3 ... fanciest version of same message goes here ... --boundary42--

From: Nathaniel Borenstein <nsb@bellcore.com>

Date: Mon, 22 Mar 1993 09:41:09 -0800 (PST)

To: Ned Freed < ned@innosoft.com >

All Body parts refer to the same data but in different formats with increasing complexiety.

HttpServletRequest는 파일 업로드를 지원안함

- HttpServletRequest는 웹 클라이언트가 전달하는 Multipart데이터를 쉽게 처리하는 메소드를 제공하지 않는다.
- 서블릿에서 파일 업로드를 처리하려면 별도의 라이브러리를 사용해야 한다. 대표적인 라이브러리가 아파치 재단의 commons-fileupload이다.

Spring MVC에서의 파일 업로드

- Spring MVC에서 파일을 업로드 하려면 몇가지 라이브러리와 설정을 추가해야 한다.
 - commons-fileupload, commons-io 라이브러리 추가
 - MultipartResolver Bean 추가

파일 업로드를 위한 라이브러리 추가

```
\(\dependency\)
      \(\rangle groupId \rangle commons-fileupload \rangle / groupId \rangle \)
      ⟨artifactId⟩commons-fileupload⟨/artifactId⟩
      ⟨version⟩1.2.1⟨/version⟩
⟨/dependency⟩
⟨dependency⟩
      \(\langle groupId \rangle commons-io \langle / groupId \rangle \)
      \(\artifactId\)\commons-io\(\artifactId\)
      ⟨version⟩1.4⟨/version⟩
</dependency>
```

스프링 설정 추가

• DispathcerServlet은 준비과정에서 "multipart/form-data" 가 요청으로 올 경우 MultipartResolver를 사용 한다.

```
@Bean
public MultipartResolver multipartResolver() {
    org.springframework.web.multipart.commons.CommonsMultipartResolver multipartResolver = new
    org.springframework.web.multipart.commons.CommonsMultipartResolver();
    multipartResolver.setMaxUploadSize(10485760); // 1024 * 1024 * 10
    return multipartResolver;
```

파일 업로드 폼

• 파일 업로드시에는 form태그에 endtype설정이 되어 있어야 한다.

Controller에서의 업로드 처리

- @PostMapping이 사용되야 한다.
- 업로드 파일이 하나일 경우 @RequestParam("file") MultipartFile file
- 업로드 파일이 여러개일 경우 @RequestParam("file") MultipartFile[] files
- MultipartFile의 메소드를 이용해서 파일이름, 파일크기 등을 구하고 InputStream을 얻어 파일을 서버에 저장한다.

Controller에서의 다운로드 처리

• 파일 다운로드와 관련된 헤더 정보를 출력한다.

```
response.setHeader("Content-Disposition", "attachment; filename=\footnotem" + fileName + "\footnotem";");
response.setHeader("Content-Transfer-Encoding", "binary");
response.setHeader("Content-Type", contentType);
response.setHeader("Content-Length", fileLength;
response.setHeader("Pragma", "no-cache;");
response.setHeader("Expires", "-1;");
```

• 파일을 읽어 HttpServletResponse의 OutputStream으로 출력한다.